Standard Classes and Interfaces -— Supplemental Reference

The following classes, interfaces, and methods are used on the UIL written exam.

class java.lang.Object

- `boolean equals(Object other)`
- `String toString()`
- `int hashCode()`

interface java.lang.Comparable

- `int compareTo(T other)`
  `// return value < 0 if this is less than other`
  `// return value = 0 if this is equal to other`
  `// return value > 0 if this is greater than other`

class java.lang.Integer implements java.lang.Comparable

- `Integer(int value)`
  `// constructor`
- `int intValue()`
- `boolean equals(Object other)`
- `String toString()`
- `int compareTo(Integer anotherInteger)`
  `// specified by java.lang.Comparable`
- `static int parseInt(String s)`
  `// Parses the string argument as a signed decimal integer.`

class java.lang.Double implements java.lang.Comparable

- `Double(double value)`
  `// constructor`
- `double doubleValue()`
- `boolean equals(Object other)`
- `String toString()`
- `int compareTo(Double anotherDouble)`
  `// specified by java.lang.Comparable`

class java.lang.String implements java.lang.Comparable

- `int compareTo(String anotherString)`
  `// specified by java.lang.Comparable`
- `boolean equals(Object other)`
- `int length()`
- `String substring(int from, int to)`
  `// returns the substring beginning at from`
  `// and ending at to-1`
- `String substring(int from)`
  `//returns substring(from, length())`
- `int indexOf(String s)`
  `// returns the index of the first occurrence of s;`
  `// returns -1 if not found`
- `int indexOf(String str, int fromindex)`
  `// Returns the index within this string of the first occurrence`
  `// of the specified substring, starting the`
  `// search at the specified index.`
- `char charAt(int index)`
  `// Returns the character at the specified index.`
- `int indexOf(int ch)`
  `// Returns the index within this string of the first occurrence`
  `// of the specified character.`
- `int indexOf(int ch, int fromindex)`
  `// Returns the index within this string of the first occurrence`
  `// of the specified character, starting the`
  `// search at the specified index.`
- `String toLowerCase()`
  `// Converts all of the characters in this String to lower`
  `// case using the rules of the default locale.`
- `String toUpperCase()`
  `// Converts all of the characters in this String to upper`
  `// case using the rules of the default locale.`
- `String[] split(String regex)`
  `// Splits this string around matches of the given regular`
  `// expression.`

class java.lang.Character

- `static boolean isDigit(char ch)`
- `static boolean isLetter(char ch)`
- `static boolean isLetterOrDigit(char ch)`
- `static boolean isLowerCase(char ch)`
- `static boolean isUpperCase(char ch)`
- `static char toUpperCase(char ch)`
- `static char toLowerCase(char ch)`

class java.lang.Math

- `static int abs(int x)`
- `static double abs(double x)`
- `static double pow(double base,`
  `double exponent)`
- `static double sqrt(double x)`
- `static double ceil(double a)`
- `static double floor(double a)`
- `static double min(double a, double b)`
- `static double max(double a, double b)`
- `static int min(int a, in b)`
- `static int max(int a, int b)`
- `static long round(double a)`

class java.util.Random

- `int nextInt()`
- `double nextDouble()`

interface java.util.List<E>

- `boolean add(E x)`
- `int size()`
- `Iterator<E> iterator()`
- `ListIterator<E> listIterator()`

class java.util.ArrayList<E> implements java.util.List<E>

- Methods in addition to the `List` methods:
- `E get(int index)`
- `E set(int index, E x)`
  `// replaces the element at index with x`
- `void add(int index, E x)`
  `// inserts x at position index, sliding elements`
  `// at position index and higher to the right`
  `// (adds 1 to their indices) and adjusts size`
- `E remove(int index)`
  `// removes element from position index, sliding elements`
  `// at position index + 1 and higher to the left`
  `// (subtracts 1 from their indices) and adjusts size`

class java.util.LinkedList<E> implements java.util.List<E>

- Methods in addition to the `List` methods
- `void addFirst(E x)`
- `void addLast(E x)`
- `E getFirst()`
- `E getLast()`
- `E removeFirst()`
- `E removeLast()`

interface java.util.Set<E>

- `boolean add(E x)`
- `boolean contains(Object x)`
- `boolean remove(Object x)`
- `int size()`
- `Iterator<E> iterator()`

class java.util.HashSet<E> implements java.util.Set<E>
class java.util.TreeSet<E> implements java.util.Set<E>

interface java.util.Map<K,V>

- `Object put(K key, V value)`
- `V get(Object key)`
- `boolean containsKey(Object key)`
- `int size()`
- `Set<K> keySet()`
- `Set<Map.Entry<K, V>> entrySet()`

class java.util.HashMap<K,V> implements java.util.Map<K,V>
class java.util.TreeMap<K,V> implements java.util.Map<K,V>

interface java.util.Map.Entry<K,V>

- `K getKey()`
- `V getValue()`
- `V setValue(V value)`

interface java.util.Iterator<E>

- `boolean hasNext()`
- `E next()`
- `void remove()`

interface java.util.ListIterator<E> extends java.util.Iterator<E>

- Methods in addition to the `Iterator` methods
- `void add(E x)`
- `void set(E x)`

class java.lang.StringBuffer

- `StringBuffer append(char c)`
- `StringBuffer append(String str)`
- `StringBuffer append(StringBuffer sb)`
- `int capacity()`
- `char charAt(int index)`
- `StringBuffer delete(int start, int end)`
- `StringBuffer deleteCharAt(int index)`
- `StringBuffer insert(int offset, char c)`
- `StringBuffer insert(int offset, String S)`
- `int length()`
- `void setCharAt(int index, char ch)`
- `String substring(int start)`
- `String substring(int start, int end)`
- `String toString()`

class java.lang.Exception

- `Exception()`
- `Exception(String message)`

class java.util.Scanner

- `Scanner(InputStream source)`
- `boolean hasNext()`
- `boolean hasNextInt()`
- `boolean hasNextDouble()`
- `String next()`
- `int nextInt()`
- `double nextDouble()`
- `String nextLine()`
- `Scanner useDelimiter(String pattern)`
  `// Sets this scanner's delimiting pattern to a pattern`
  `// constructed from the specified`
  `// String.`